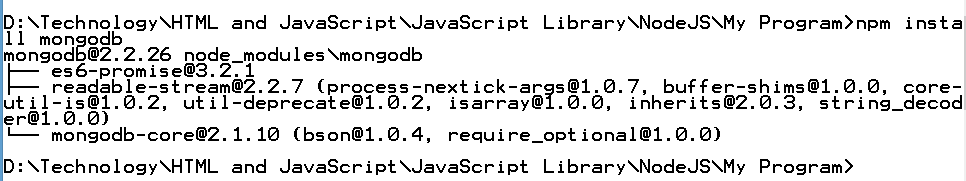
**Mongo DB with Node JS**

Install Node.js and npm

To install the Mongo db using the npm

**npm install mongodb**



Declare MongoClient variable and other variables

var mongoClient = require("mongodb").MongoClient;

var assert = require("assert");

Connect using the mongoClient to a running mongod instance by specifying.

For Example, the following code connects to a MongoDb instance that run on the localhost interface on port 27017 and switch to the mydb database.

**MongoTest.js**

var mongoClient = require("mongodb").MongoClient;

var assert = require("assert");

var url = "mongodb://localhost:27017/mydb";

mongoClient.connect(url,function(err,db){

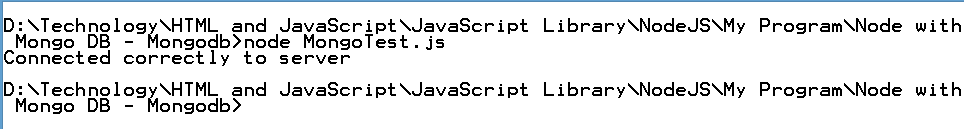
assert.equal(null,err);

console.log("Connected correctly to server");

db.close();

});

After running this program



**Insert Data with Node.js**

We can use the insertOne method and the insertMany method to add documents to a collection in MongoDb. If we attempt to add documents to a collection that does not exist. MongoDb will create the collection for you.

**InsertOne.js**

var mongoClient = require("mongodb").MongoClient;

var assert = require("assert");

var objectId = require("mongodb").ObjectId;

var url= "mongodb://localhost:27017/mydb";

var insertDoc = function(db,callback){

db.collection("Employee").insertOne({\_id:102,Name:"Seeta",Salary:14000},function(err,result){

assert.equal(err,null);

console.log("record inserted successfully");

callback();

});

};

mongoClient.connect(url,function(err,db){

assert.equal(null,err);

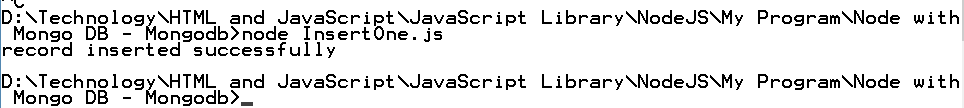
insertDoc(db,function(){

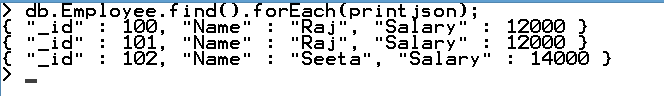
db.close();

});

});

After running this program





**Find or Query Data with Node.js**

We can use the find() method or issue a query to retrieve data from a collection in MondoDB. All queries in MongoDB have the scope of a single collection.

Queries can return all documents in a collection or only the documents that match a specified filter or criteria. we can specify the filter or criteria in a document and pass as a parameter to find method.

The find() method return query result in a cursor, which is an iterable object that yields documents.

**Query for All documents in a Collection**

To return all documents in a collection, call the find() method without a criteria document. For example, the following operations queries for all documents in the Employee collection.

**RetrieveQuery.js**

var mongoClient = require("mongodb").MongoClient;

var assert = require("assert");

var objectId = require("mongodb").ObjectId;

var url= "mongodb://localhost:27017/mydb";

var retrieveQuery = function(db,callback){

var cursor = db.collection("Employee").find();

cursor.each(function(err,doc){

assert.equal(err,null);

if(doc!=null) {

console.log(doc);

}else {

callback();

}

});

};

mongoClient.connect(url,function(err,db){

assert.equal(null,err);

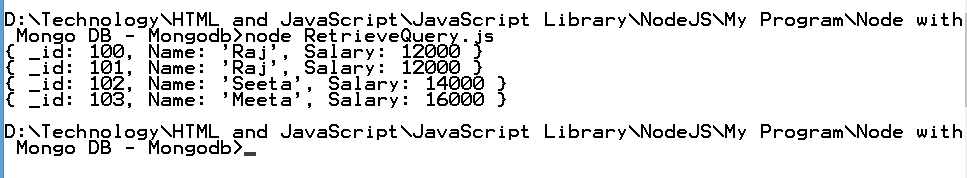
retrieveQuery(db,function(){

db.close();

});

});

After running this program



**RetrieveWithCondition.js**

var mongoClient = require("mongodb").MongoClient;

var assert = require("assert");

var objectId = require("mongodb").ObjectId;

var url= "mongodb://localhost:27017/mydb";

var retrieveQuery = function(db,callback){

var cursor = db.collection("Employee").find({Name:"Raj"});

cursor.each(function(err,doc){

assert.equal(err,null);

if(doc!=null) {

console.log(doc);

}else {

callback();

}

});

};

mongoClient.connect(url,function(err,db){

assert.equal(null,err);

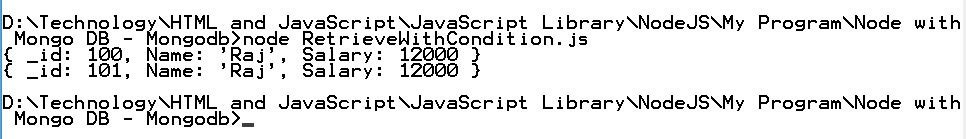
retrieveQuery(db,function(){

db.close();

});

});

After running this program



**RetrieveWithConditio.js**

var mongoClient = require("mongodb").MongoClient;

var assert = require("assert");

var objectId = require("mongodb").ObjectId;

var url= "mongodb://localhost:27017/mydb";

var retrieveQuery = function(db,callback){

//var cursor = db.collection("Employee").find({Name:"Raj"});

var cursor = db.collection("Employee").find({Salary:{$gt:13000}});

cursor.each(function(err,doc){

assert.equal(err,null);

if(doc!=null) {

console.log(doc);

}else {

callback();

}

});

};

mongoClient.connect(url,function(err,db){

assert.equal(null,err);

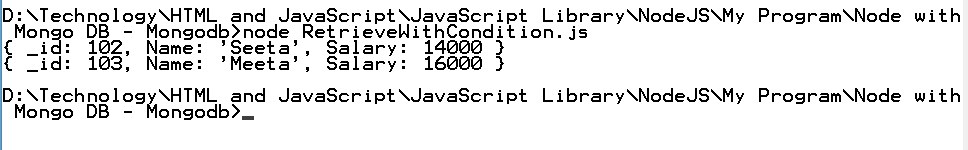
retrieveQuery(db,function(){

db.close();

});

});

After running this program



**Sort the Query**

var mongoClient = require("mongodb").MongoClient;

var assert = require("assert");

var objectId = require("mongodb").ObjectId;

var url= "mongodb://localhost:27017/mydb";

var retrieveQuery = function(db,callback){

//var cursor = db.collection("Employee").find({Name:"Raj"});

var cursor = db.collection("Employee").find().sort({Salary:-1});

cursor.each(function(err,doc){

assert.equal(err,null);

if(doc!=null) {

console.log(doc);

}else {

callback();

}

});

};

mongoClient.connect(url,function(err,db){

assert.equal(null,err);

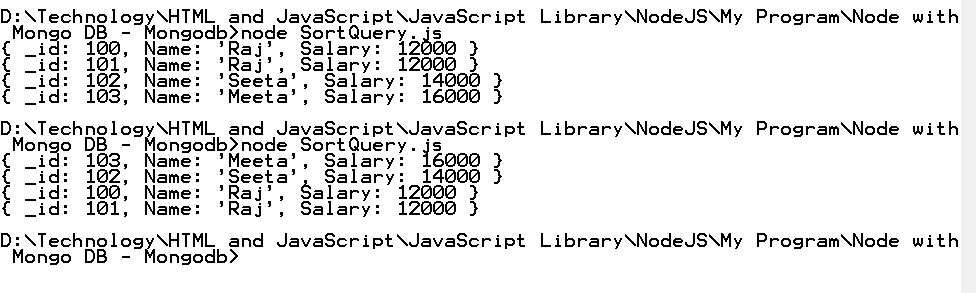
retrieveQuery(db,function(){

db.close();

});

});

After running this program



**Update Data with Node.js**

You can use the UpdateOne() method, updateMany() method, and replaceOne() method to update documents of a collection.